Learning and optimization of an aspect hidden Markov model for query language model generation

Conference or Workshop Item

LEARNING AND OPTIMIZATION OF AN ASPECT HIDDEN MARKOV MODEL FOR QUERY LANGUAGE MODEL GENERATION

Qiang Huang, Dawei Song, Stefan Rüger
Knowledge Media Institute, The Open University, Milton Keynes, UK
{*q.huang d.song s.rueger*}*@open.ac.uk*

Peter Bruza
School of Information Systems, Queensland University of Technology, Australia
p.bruza@qut.edu.au

Keywords: Aspect Model, Latent Variable Model, Segmentation, Information Retrieval.

Abstract: The Relevance Model (RM) incorporates pseudo relevance feedback to derive query language model and has shown a good performance. Generally, it is based on uni-gram models of individual feedback documents from which query terms are sampled independently. In this paper, we present a new method to build the query model with latent state machine (LSM) which captures the inherent term dependencies within the query and the term dependencies between query and documents. Our method firstly splits the query into subsets of query terms (i.e., not only single terms, but different combinations of multiple query terms). Secondly, these query term combinations are then considered as weighted latent states of a hidden Markov Model to derive a new query model from the pseudo relevant documents. Thirdly, our method integrates the Aspect Model (AM) with the EM algorithm to estimate the parameters involved in the model. Specifically, the pseudo relevant documents are segmented into chunks, and different chunks are associated with different weights in relation to a latent state. Our approach is empirically evaluated on three TREC collections, and demonstrates statistically significant improvements over a baseline language model and the Relevance Model.

1 INTRODUCTION

1.1 Background

The Relevance Model (RM) (Laverenko and Croft, 2001) has been regarded as a promising the language modeling approach to document retrieval. From a practical point of view variants of the RM have shown encouraging performance in adhoc search (Laverenko and Croft, 2001), cross-language retrieval (Laverenko et al., 2002) and topic detection and tracking (Lavrenko et al., 2002).

The relevance model computes $P(w|R)$ which is interpreted as the probability of observing a word w in documents relevant to an information need (R). In practice, it is approximated by $P(w|Q)$ for a query Q. Computing this probability for every term w in the vocabulary yields an estimate of the true relevance model. In the RM, the query Q is considered to be a random sample from the unknown relevance model R. R can be envisaged as an unknown process from which words can be sampled. then the question is: if query terms $q_1, \ldots, q_m$ have been sampled, what is the probability of term w will be sampled next. Essentially this probability can be expressed in term of the probability of co-occurrence between w and Q, which is estimated by sampling the query terms from w via a number of unigram distributions M_i. Operationally, the top ranked documents retrieved by the query Q are used to serve as these distributions. Of particular importance to this article is the query terms are sampled independently of each other (Laverenko and Croft, 2001).

Intuitively, however, there often exist dependencies between query terms. In addition, the combinations of query terms often carry more information than single terms individually. Furthermore, the distributions $P(M_i)$ should not be kept uniform, but should depend on the query terms and w. Our hypothesis is that incorporating these factors into the Relevance Model will improve the retrieval effectiveness. In this paper, we propose to use the idea of the latent variable model (LVM) to establish connections between a document D and a term w by detecting

dependencies between the document and latent variables, and to use the aspect model to capture the different contributions of different chunks of documents to the relevance.

1.2 Related work

In statistics, latent variables are variables that are not directly observed but are rather inferred from other variables that are observed and directly measured. Modeling the observed text as generated from latent aspects or topics is a prominent approach in machine learning (Gruber et al., 2007). In recent years, latent variable models (LVM) have been widely applied to information retrieval and natural language processing.

Hidden Markov Model (HMM), as a type of LVM, has been applied to passage retrieval, which returns relevant passages instead of whole documents (Jiang and Zhai, 2006), and topic segmentation, which segments a document and extracts the topic-related contents (Blei and Moreno, 2001). The former (Jiang and Zhai, 2006) builds a two-state like structure to look for the topical boundaries between relevant and irrelevant passages given a specific query. The latter (Blei and Moreno, 2001) depends on a multiple-state HMM to lable the contents in the document collections without taking into account a specific query. Additionally, in (Blei and Moreno, 2001), the combination of an aspect model and a HMM can also generate the observation probabilities for new segmentations.

In (Gildea and Hofmann, 1999), for on-line word prediction, a statistical language model is used to capture topic-related long-range dependencies, in which the topics are modelled in a latent variable framework. Different from the method (Bellegarda, 1997) which is based on Latent Semantic Analysis (LSA), this language model leads to a new application on probabilistic Latent Semantic Analysis (pLSA).

Probabilistic Latent Semantic Indexing (pLSI) (Hofmann, 1999) and Latent Dirichlet Allocation (LDA) (Blei et al., 2003) are two popular probabilistic text modeling techniques. As a classical LVM, pLSA was designed as discrete counterpart of LSA to provide a better fit to text data, which models general co-occurrence data associated with an unobserved class (topic) variable with each observation (Wei and Croft, 2006). Additionally, since it is based on the likelihood principle, it can not only define a generative data model and directly minimize word perplexity, but also take advantage of statistical methods for model fitting and model combination. However, because of the large number of documents, the model is prone to overfitting. Furthermore, since each training document has its own set of topic weighting parameters, pLSA does not provide a generative framework for describing the probability of an unseen document (Hsu and Glass, 2005). In order to address these shortcomings, the LDA model (Blei et al., 2003) introduces a more consistent probabilistic approach as it ties the parameters of all documents via a hierarchy generative model. Generally, LDA treats a document as a mixture of multiple topics, which are generated from a Dirichlet prior mutual to all documents in the corpus. With the number of model parameters dependent only on the number of topic mixtures and vocabulary size, LDA is less prone to overfitting and is capable of estimating the probability of unobserved test documents.

The latent topics are often assumed independent of each other, given words in documents. However, this independent assumption gives too few constraints on the distribution of the latent "topics" and the observed words, thus affecting the precision of the parameter estimation. Recent research have tried to add some specific conditions on the model. In (Gruber et al., 2007), the independent assumption is dropped by, for example, assuming that all the words in the same sentence are about the same topic, and successive sentences are more likely to be about the same topic. In (Wang and McCallum, 2005), a topical $N-$gram model is used to automatically form a n-gram from its surrounding context by considering the order of words.

1.3 Our approach

Compared with the application of LVM, such as LDA and pLSA, to document models, the application of LVM to query models has yet to be explored. In this paper, we propose to further relax the independence assumption and present a novel framework based on the aspect hidden Markov model. Specifically we assume that "latent topics" are governed by a Markov model and apply the aspect model for enhance the learning of prior distribution of topics. We use single query terms as well as combinations of query terms as the latent states. We have developed an innovative method to learn and optimize the inter-state dependencies, i.e., the so-called high-order term relationships (e.g. (Java + programming)$\rightarrow$ computer) from the pseudo relevance feedback documents which are divided into chunks. Distinct from other term association deviation approaches, we assume that different chunks of documents are assigned different weights automatically adjustable from Aspect Model (AM). The AM will run iteratively to optimize its parameters.

The remainder of this paper is organized as fol-

lows. Section 2 lays out the basic theory of the Aspect Model and gives the EM algorithm for the estimation of parameters; Section 3 describes the theory of association rule and the application of smoothing; Section 4 presents experimental results on three TREC ad-hoc collections, and compares with Query Likelihood (QL) based language model and the Relevance Model; Section 5 concludes the paper and highlights future research directions.

2 Theoretical Framework

2.1 The Aspect Model

In this section some brief descriptions of the aspect model, which was first used by Hofmaan and Puzicha (Hofmann and Puzicha, 1999), will be presented.

The aspect model is a latent variable model for co-occurrence data. Given documents $D \in \mathbf{D} = \{D_1, D_2, \ldots, D_N\}$, and the terms w from a vocabulary $\mathbf{V}$, i.e. $w \in \mathbf{V} = \{w_1, \ldots, w_M\}$ that they contain, an observation (D, w) ia associated with a latent variable $S \in \mathbf{S} = \{S_1, \ldots, S_K\}$. Conceptually, the latent variables are topics embedded in the document collection. One can think of a process where documents generate or "induce" the topics or latent classes, which in turn generate terms according to class specific distributions (Schein et al., 2001). Documents are assumed to be independent of terms, given the topics. The joint probability distribution over documents, topics, and terms is (Schein et al., 2001):

$$P(D, w, S) = P(S)P(D|S)P(w|S) \tag{1}$$

Assuming that S are exhaustive and mutually exclusive, we can sum over the possible values of S when calculating the joint distribution of a document and a term:

$$P(D, w) = \sum_S P(S)P(D|S)P(w|S) \tag{2}$$

The parameters in Equation 2 are explained as follows. $P(w|S)$ can be viewed as a language model of latent variable S. $P(D|S)$ is a probability distribution over the training documents. $P(S)$ is the prior distribution on S.

In our application, we prefer using segments of a document rather than the whole document. The motivation is that the different parts (e.g., a sentence, or text within a window) of a document may have different contributions to the aspect model. Let d denote a segment in collection $\mathbf{d} = \{d_1, \ldots, d_N\}$ of pre-segmented documents, w denote a term, and S denote a latent topic.

Given a corpus of N document segments and the words within those segments (w_n^d), the training data for an aspect model is the set of pairs $\{(d_n, w_n^d)\}$ for each segment label and each term in those segments. The Expectation Maximization (EM) algorithm can be used to fit the parameters of Equation 2 from an un-categorized corpus. This corresponds to learning the underlying topics of a corpus $P(w|S)$ as well as the degree to which each training document is about those topics $P(d|S)$ (Blei and Moreno, 2001).

In the *E-step*, we compute the posterior probability of the hidden variable given our current model.

E-step:

$$P(S|d, w) = \frac{P(S)P(d|S)P(w|S)}{\sum_{S'} P(S')P(d|S')P(w|S')} \tag{3}$$

In the *M-step*, we maximize the log likelihood of the training data with respect to the parameters $P(S)$, $P(d|S)$ and $P(w|S)$.

M-step:

$$P(d|S) = \frac{\sum_{w \in V} P(S|d, w)n(d, w)}{\sum_{w \in V} \sum_{d' \in \mathbf{d}} P(S|d', w)n(d', w)} \tag{4}$$

$$P(w|S) = \frac{\sum_{d \in \mathbf{d}} P(S|d, w)n(d, w)}{\sum_{w' \in V} \sum_{d \in \mathbf{d}} P(S|d, w')n(d, w')} \tag{5}$$

$$P(S) = \frac{\sum_{d \in \mathbf{d}} \sum_{w \in V} P(S|d, w)n(d, w)}{\sum_{S'} \sum_{w \in V} \sum_{d \in \mathbf{d}} P(S'|d, w)n(d', w)} \tag{6}$$

where $n(d, w)$ is the number of times that the term w appears in the segment d. A detailed discussion can be found in (Hofmann, 1999).

2.2 Derivation of Our Theory

As introduced in Section 1, we propose to divide a query into all the combinations of query terms, for example, the original query $\mathbf{Q}_o = \{q_1, q_2\}$ is pre-expanded to $\mathbf{Q} = \{\{q_1\}, \{q_2\}, \{q_1, q_2\}\}$. Here, Q_j is used to represent each new element (i.e., a subset of the query terms) in the pre-expanded query, formally $Q_j \in \mathbf{Q} = \{Q_1, Q_2, \cdots, Q_M\}$. The intuition is that different combinations (subsets) of query terms all contribute but in different degrees to the query model. Therefore, not only individual terms (as in the Relevance Model), but also all these combinations should be considered in the query model derivation process. An example of the query pre-expansion is shown in figure 1:

Consequently, in the process of computing the probability, $P(w|\mathbf{Q})$, of generating a term w in the segment d given a query Q, the contributions from all the query terms is collected including contributions from

$$\begin{array}{ll} \textit{Query}: & \{aspect, model\} \\ \textit{ExpandQuery}: & \{\{aspect\}, \{model\}, \{aspect + model\}\} \end{array}$$

Figure 1: Example of the query terms combination.

single query terms or combinations of query terms (denoted Q_j):

$$P(w|\mathbf{Q}) = \sum_{Q_j \in \mathbf{Q}, d \in \mathbf{d}} P(w|Q_j, d) P(d|Q_j) P(Q_j|\mathbf{Q}) \quad (7)$$

where $P(w|Q_j, d)$ represents the probability of the term w being generated given a subset of query terms Q_j and a segment d; $P(d|Q_j)$ is the probability distribution over segments given Q_j; and $P(Q_j|\mathbf{Q})$ is the prior distribution of query subset; $\mathbf{d}$ represents the collection of segments in the feedback documents.

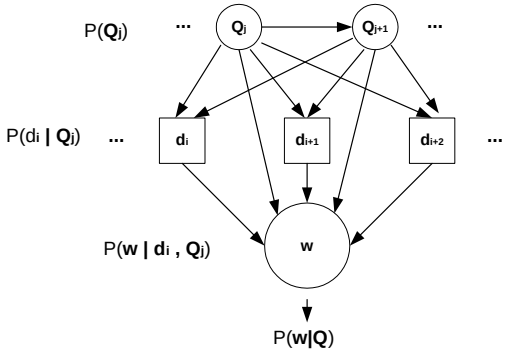

Figure 2: Induction of Structure.

Figure 2 shows an example of the relationships between the subsets of query terms (Q_j), the segments (d_i) of feedback documents and vocabulary terms (w) occurring in feedback documents , which is a Bayesian network like structure. Instead of assumming the independent between different Q_j, they are governed by a Markov model. Although these Q_j are not really "hidden", in order to estimate the parameters of system, we consider them as "latent" variables which act as "hidden" states in hidden Markov model (HMM). More specifically, each Q_j is considered as a "hidden" state of the HMM, and represented as $S_{Q_j} \in \mathbf{S}_Q = \{S_{Q_1}, S_{Q_2}, \ldots, S_{Q_M}\}$, which corresponds to $Q_j \in \mathbf{Q} = \{Q_1, Q_2, \cdots, Q_M\}$..

The aspect hidden Markov model (AHMM) (Blei and Moreno, 2001) is applied to estimate the probability distribution of each state S_{Q_j} and transition probabilities. The aspect model associates an unobserved class variable with each observation (Hofmann, 1999). The aspect model is embedded in a HMM, which can determine both the observation emission probabilities and training segment clusters to find the transition probabilities. The detailed application of AHMM in our approach will be explained in the following section.

2.3 Model Learning and Optimization

The AHMM (Blei and Moreno, 2001) segments a new document by dividing its words into observation windows of size L and running the Viterbi algorithm to find the most likely sequence of "hidden" state which generate the given document. For optimization, the parameters of the model need to be estimated iteratively. As shown in Section 2.1, the Aspect Model provides an effective way to estimate parameters. Considering the application of the model for the retrieval task, the same basic approach to estimation is taken but on a different tack:

Pre-segmentation: A number F of feedback documents is used as basis. In the experiments reported in the next section, documents are segmented into chunks using fixed-length window of W words with O words overlapping. Here, each segmented chunk is treated as a "new" document.

Construction of States: As mentioned in Section 2.2, the combinations of query terms are considered, i.e. Q_j, as "hidden" states, i.e. S_{Q_j}.

Initial Clustering: In order to build initial clusters corresponding to each state S_{Q_j}, only those chunks including the query terms in Q_j are selected. These chunks serve as initial training set. The chunks not containing any query terms will be checked for which state they belong to in the next steps.

Training Aspect Model: In the process of training, Equations $3 \sim 6$ are used to estimate the parameters, i.e., $P(d|S_{Q_j})$, $P(w|S_{Q_j})$ and $P(S_{Q_j})$, of the model with the clustered chunks. Due to of the problem of data sparsity, the model is run once.

Label other chunks: The top L words related to Q_j have been computed according to the value of $P(w|S_{Q_j})$ are used as the basis of vector space to represent State S_{Q_j}. In order to label the chunks not containing query terms, the conditional probability $P(d|S_{Q_j})$ needs to be estimated as follows:

Let $d^k = \{w_1, w_2, w_3, \ldots, w_k\}$ denote a segment of k terms, and $d^W = d$ denotes the full observation with W words in a window. After obtaining $P(w|S_{Q_j})$ and $P(d|S_{Q_j})$, an approximation to EM (Blei and Moreno, 2001) is applied to find $P(d|S_{Q_j})$:

$$P(d|S_{Q_j}) = \frac{P(S_{Q_j}|d)P(d)}{P(S_{Q_j})} \quad (8)$$

in which $P(S_{Q_j}|d)$ is approximated recursively as follows:

$$P(S_{Q_j}|d^k) = \frac{1}{k+1}\frac{P(w_k|S_{Q_j})P(S_{Q_j}|d^{k-1})}{\sum_{S'_{Q_j}} P(w_k|S'_{Q_j})P(S'_{Q_j}|d^{k-1})} + \frac{k}{k+1}P(S_{Q_j}|d^{k-1}) \quad (9)$$

where $P(S_{Q_j}|d^0) = P(w_1|S_{Q_j})$.

Then, we label the chunk with S_{Q_j} maximizing $P(S_{Q_j}|d)$:

$$S^*_{Q_j} = \arg\max_{S_{Q_j}} P(S_{Q_j}|d) \quad (10)$$

Update $P(S_{Q_j})$ After labelling the remaining chunks, it is necessary to update the $P(S_{Q_j})$:

$$\begin{aligned} &for\ j = 1 : |\mathbf{S}_Q|, \\ &\quad Num_{S_{Q_j}} = Orignial\_Num_{S_{Q_j}} + \\ &\qquad Labelled\_Num_{S_{Q_j}}; \\ &\quad P(S_{Q_j}) = Num_{S_{Q_j}}/|Chunks|; \\ &end \end{aligned}$$

Here, $|\mathbf{S}_Q|$ signifies the total number of states, $Orignial\_Num_{S_{Q_j}}$ is the number of chunks in the initial cluster, $Labelled\_Num_{S_{Q_j}}$ is the number of chunks which are labelled with S_{Q_j} in previous step, and $|Chunks|$ is the total number of chunks.

Re-estimation of parameters of Apect Model: Up till this point, $P(S_{Q_j})$ has been updated and all chunks with different states have been labelled. This step is to iterate the previous steps (from the "Training Aspect Model" step) to re-estimate the parameters of the Aspect Model.

2.4 Query Model Generation

By running the model learning and optimization process (Section 2.3), the vocabulary terms, i.e., w, are ranked and selected according to the conditional probabilities $P(w|\mathbf{Q})$ defined in the Equation 7. The $P(w|Q_j,d)$, $P(d|Q_j)$ and $P(Q_j|\mathbf{Q})$ in Equation 7 are approximated by the estimates $P(w|S_{Q_j})$, $P(d|S_{Q_j})$, and $P(S_{Q_j})$.

Given the original query $\mathbf{Q}_o = \{q_1, \cdots, q_{|\mathbf{Q}_o|}\}$, the original query model, $P(q_i|\mathbf{Q}_o)$ is computed as:

$$P(q_i|\mathbf{Q}_o) = \frac{QTF * IDF(q_i)}{\sum_{j\in 1\cdots|\mathbf{Q}_o|} QTF * IDF(q_j)} \quad (11)$$

To build a new query model $P(w|\mathbf{Q}_s)$, the distribution $P(w|\mathbf{Q})$ is then combined with the original query model $P(q_i|\mathbf{Q}_o)$ via smoothing, a commonly used technique to combine different models, or term distributions.

Typically, linear mixture, a classical smoothing method, can be used to derive the "new" smoothed model $P(w|\mathbf{Q}_s)$:

$$P(w|\mathbf{Q}_s) = \lambda P(w|\mathbf{Q}) + (1-\lambda)P(w|\mathbf{Q}_o) \quad (12)$$

where $P(w|Q_o) = 0$ when the term w does not occur in the original query.

3 Experimental Setup

3.1 Data

The experiments are based on standard TREC data sets, including the Associated Press Newswire (AP) 1988-90 with topics 51-150, Wall Street Journal (WSJ) 1987-92 with topics 51-100 and 151-200 , San Jose Mercury News (SJMN) 1991 with topics 51-150. Only the title fields of the topics are used. Topics with no relevant documents for a specific collection have been removed from the query set. The statistics of the the collections and query sets are given in Table 1

3.2 Baselines

Two baselines were selected as benchmarks for comparison: The basic Language Model and the Relevance Model (Laverenko and Croft, 2001). The basic Language Model is an uni-gram langauge model, using Query Likelihood (QL) with Kullback-Leibler Divergence. It is built over the vocabulary, in which the likelihood of query terms is the probability of distribution over the documents collection being used. Additionally, the Relevance Model features two methods. In the experiments presented here, Method 1 was employed as a baseline, i.e. the independent identical distribution (i.i.d.) sampling to compute the relevance model (RM). Average precision is used as a measure for performance comparison.

3.3 Parameter Settings

The top 30 matched documents is used as a set of pseudo-relevant documents (i.e. $F = 30$), each of which is segmented into chunks by 30-word-length sliding window with 25 words overlapping. After

Table 1: Test collection and test topics.

Collection	Contents	# of docs	Size	Queries (topics)	# of Queies with Relevant Docs
AP	Associated Press Newswire (1988-90)	242,918	0.73Gb	51-150	99
WSJ	Wall Street Journal (1987-92)	173,252	0.51Gb	51-100 & 151-200	100
SJM	San Jose Mercury News (1991)	90,257	0.29Gb	51-150	94

Table 2: Sample probabilities from the query "the US. control of insider trading" on collection AP88-90 based aspect model.

Query : the US. control of Insider Trading							
	inside	trad	control	inside trad	inside control	trad control	control inside trad
Selected words	$P(w\|Q_1)$	$P(w\|Q_2)$	$P(w\|Q_3)$	$P(w\|Q_4)$	$P(w\|Q_5)$	$P(w\|Q_6)$	$P(w\|Q_7)$
trad	0.7534	0.3084	0.4167	0.6959	0.7484	0.3633	0.7328
inside	0.2547	0.6729	0.2863	0.3315	0.2720	0.6275	0.4531
secure	0.2103	0.2215	0.2025	0.2190	0.2249	0.2353	0.3115
stock	0.1897	0.2861	0.3116	0.2123	0.2268	0.3127	0.3017
inform	0.1416	0.1162	0.0634	0.1414	0.1380	0.1123	0.1258
sec	0.1263	0.1232	0.1616	0.1297	0.2249	0.2353	0.2367
exchange	0.0944	0.1315	0.1928	0.1042	0.1199	0.1589	0.1673
market	0.1065	0.1422	0.0409	0.1164	0.1024	0.1274	0.1345
law	0.0885	0.09	0.1908	0.0916	0.1143	0.1243	0.1277
japan	0.1096	0.1108	0.1772	0.1133	0.1305	0.1378	0.1401

learning and optimization, the top 100 terms corresponding to the original query are selected according to their estimated probability over the segmented chunks and "latent" topics. For each query, the top 100 terms with highest probability $P(w|Q)$ are selected to compute the new query model. Additionally, the interpolation coefficient λ for smoothing the query model is set to be: 0.95, 0.94 and 0.96 for AP88-90, SJM, and WSJ87-92, respectively. These parameters settings are based on past experience. Further adaptation of these parameters will be systematically investigated in future work.

4 Experiment Results and Analysis

Experimental results, using the values of parameters described in 3.3, are reported in the following tables and figures .

First the effect of different combinations of query terms to derive high-order term associations is illustrated. Consider query "*the US. control of insider trading*". After applying stemming and removing stop words, the query becomes "control inside tradt". By adding combinations of maximum 3 words, the following query is obtained: "*control*", "*inside*", "*trad*", "*control inside*", "*control trad*", "*inside trad*","*control inside trad*"

Table 2 lays out the probabilities of words generated from the AHMM model, indicating words implied by the different query states (un-normalized). The left most column lists the words occurring in the document collections, and the terms in the top row are the query terms including single query term as well as their combinations. It is not surprising the values in each row are quite variable. For example, the probability $P(trade|Q_1)$ ($Q_1 = inside$) is two times greater than the probability $P(trade|Q_2)$ ($Q_2 = trad$). The table also shows a trend that the words in the state of query term combinations hold higher probability when compared with probabilities engendered by single query term states. This accords with the intuition that longer query terms hold more information to decide the relationship between the query and the words in the documents.

The retrieval results on the *AP* collection are presented in Table 3, in which the results based on AHMM without smoothing and with smoothing are given. Additionally, the results using Query Likelihood with Kullback-Leibler Divergence (QL), and

Table 3: Comparison of Aspect Hidden Markov Model to basic language model and Relevance Model on three data sets AP88-90, SJM, and WSJ87-92.

Collection	QL	RM	AHMM (no smoothing)	AHMM (smoothing)	%chg over QL	%chg over RM
AP88-90	0.2219/10388	0.2725/12215	0.2993/12833	0.3023/12876	+36.1**	+10.9*
SJM	0.2011/3005	0.2502/3381	0.2578/3483	0.2603/3512	+29.4**	+4.0*
WSJ87-92	0.2868/6701	0.3287/7459	0.3384/7548	0.3420/7568	+19.2**	+4.0*

$**$ indicates that the difference is statistically significant according to t-test at the level of $p-value < 0.01$
$*$ indicates that the difference is statistically significant according to t-test at the level of $p-value < 0.05$

Relevance Model (RM) are also listed for comparison. When compared with QL on three collections, AHMM with smoothing shows encouraging improvement on collections AP88-90, SJM and WSJ8792 with respective increases 36.1%, 29.4% and 19.2% on average precision. T-tests reveal their statistical significance at level of $p-value < 0.01$.

Comparison with RM also shows significany performance improvements on collection AP88-90 with 10.9% increase of average precision at the ($p-value < 0.01$), and about 4% increases on collection SJM and WSJ87-92 ($p-value < 0.05$).

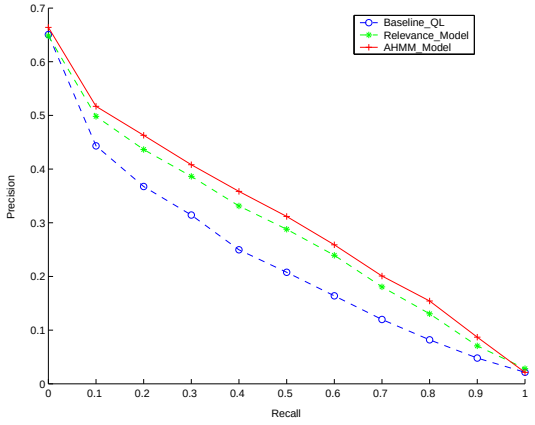

Figure 3: Retrieval performance of aspect hidden markov model on the AP88-90 dataset with Trec 51-150 title queries.

Figures 3 $\sim$ 5 show comparisons of Recall-precision curves using the three methods (QL, RM, and Aspect with smoothing) on three collections with different query topics, respectively. In Figure 3, AHMM with smoothing shows saliently higher precisions at almost all recall points than that of using QL and RM. In Figure 4 and 5 similar performances are also obtained in the comparison with QL, and the performance gap between the two curves of AHMM and RM is closer.

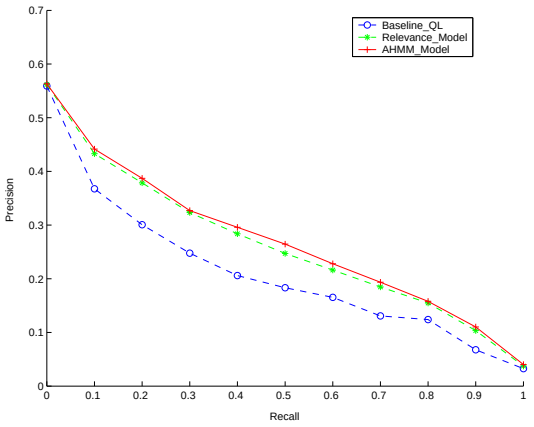

Figure 4: Retrieval performance of aspect hidden markov model on the SJM dataset with Trec 51-150 title queries.

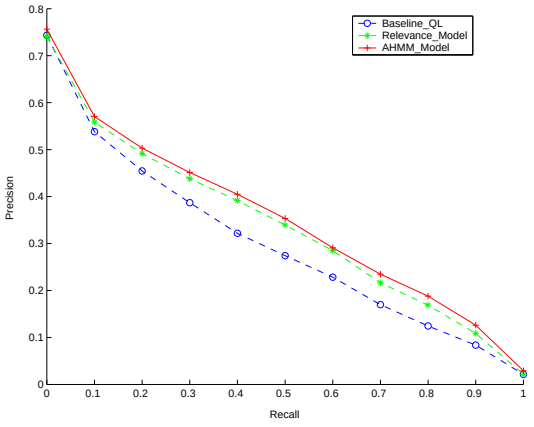

Figure 5: Retrieval performance of aspect hidden markov model on the WSJ87-92 dataset with Trec 51-100 & 151-200 title queries.

5 Conclusion and Future Work

This paper proposes an Aspect Hidden Markov Model (AHMM) for query model derivation. Firstly, the original query is expanded by different combinations of query terms. The effects of different combinations is estimated by considering them as the "hidden" states in AHMM. Secondly, documents are segmented into chunks so more relevant portions of a document can be brought into consideration. The occurrence probability of a chunk is estimated relative to different "hidden" states. Thirdly, the high-order dependency between a subset of query terms and terms in the documents are estimated by running the AHMM.

The use of AHMM can better estimate the conditional probability $P(w|S)$ of a word *w* given a specific "latent" state, and the prior probability distribution $P(S)$ of "latent" states. The estimation of $P(S)$ implies dropping the assumption that the distribution of "latent" variable is uniform. On the other hand, the pre-expansion of query by decomposing and combining query terms expands the observation space, which is more powerful than just using individual query terms. This helps capture more important relationships between query terms and words in the documents.

According to the results presented in this paper, the application of AHMM by treating the query terms as "latent states" shows encouraging performance. However, there remain aspects for further exploration. The first is the over-fitting problem when using AHMM. As the query model is trained on the 30 feedback documents, there is a problem of data sparsity. In an attempt to counter this problem, overlapping windows were employed, however over-fitting remains a challenging problem when running multiple iterations. Fixed parameters were used for window size and number of expansion terms, so more research is needed to optimally tune these parameters. Further tests are planned involving other smoothing methods rather than only using the linear interpolation. Finally, we will also compare with other term dependency based query language modelling approaches in the future.

REFERENCES

Bellegarda, J. (1997). A latent semantic analysis framework for large-span language modeling. In *Proceedings of 6th European Conference on Speech Communication and Technology*, Rhodes, Greece.

Blei, D. M. and Moreno, P. J. (2001). Topic segmentation with an aspect hidden markov model. In *Proceedings of SIGIR'01*, New Orleans, Louisiana, USA.

Blei, D. M., Ng, A. Y., and Jordan, M. J. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, pages 993–1022.

Gildea, D. and Hofmann, T. (1999). Topic-based language models using EM. In *Proceedings of 6th European Conference on Speech Communication and Technology*.

Gruber, A., Rosen-Zvi, M., and Weiss, Y. (2007). Hidden topic markov models. In *Artificial Intelligence and Statistics (AISTATS)*, San Juan, Puerto Rico.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of SIGIR'99*, Berkeley, CA, USA.

Hofmann, T. and Puzicha, J. (1999). Latent class models for collaborative filtering. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence)*, pages 688–693.

Hsu, B.-J. and Glass, J. (2005). In *Proceedings of Empirical Methods in Natual Language Proceedings*, Sydney, Austrilia.

Jiang, J. and Zhai, C. (2006). Extraction of coherent relevant passages using hidden markov models. *ACM Transactions on Information Systems*, 24(3):295–319.

Laverenko, V., Choquetto, M., and Croft, W. B. (2002). Cross-lingual language models. In *Proceedings of SIGIR'02*.

Laverenko, V. and Croft, W. B. (2001). Relevance-based language models. In *Proceedings of SIGIR'01*, pages 120–127.

Lavrenko, V., Allan, J., DeGuzman, E., LaFlamme, D., Pollard, V., and Thomas, S. (2002). Relevance models for topic detection and tracking. In *Proceedings of the Conference on Human Language Technology (HLT)*.

Schein, A. I., Popescul, A., and Ungar, L. H. (2001). Pennaspect: A two-way aspect model implementation. Technical Report MS-CIS-01-25, University of Pennsylvania.

Wang, X. and McCallum, A. (2005). A note on topical n-grams. Technical Report UM-CS-2005-071, University of Massachusetts.

Wei, X. and Croft, W. B. (2006). Lda-based document models for ad-hoc retrieval. In *Proceedings of SIGIR'06*, pages 178–185.